`rand` problems

simulation meeting—January 29, 2004

Joel Heinrich—University of Pennsylvania

We briefly summarize the contents of the recent CDF note
www-cdf.fnal.gov/cdfnotes/cdf6850_badrand.pdf

- `rand` (Linux) = `random` (UNIX/Linux) has a statistical defect
- `rand` (UNIX) only gives 15 bits—too few
- best to avoid C/C++ `rand` altogether (see "Numerical Recipes in C")
- CLHEP's `RandEngine` on Linux has a **major** bug—a victim of different ranges for `rand` on UNIX vs Linux

On Linux, C/C++ `rand` is a 31-bit generator also available as `random` on both Linux and UNIX. CDF6850 shows that `random` fails the simple "maximum spacing" test, while all "standard" generators pass the test. A famously bad generator "`randu`" also fails the test. The defect in `random` probably lies in its method, which does not seem to be discussed in standard literature.

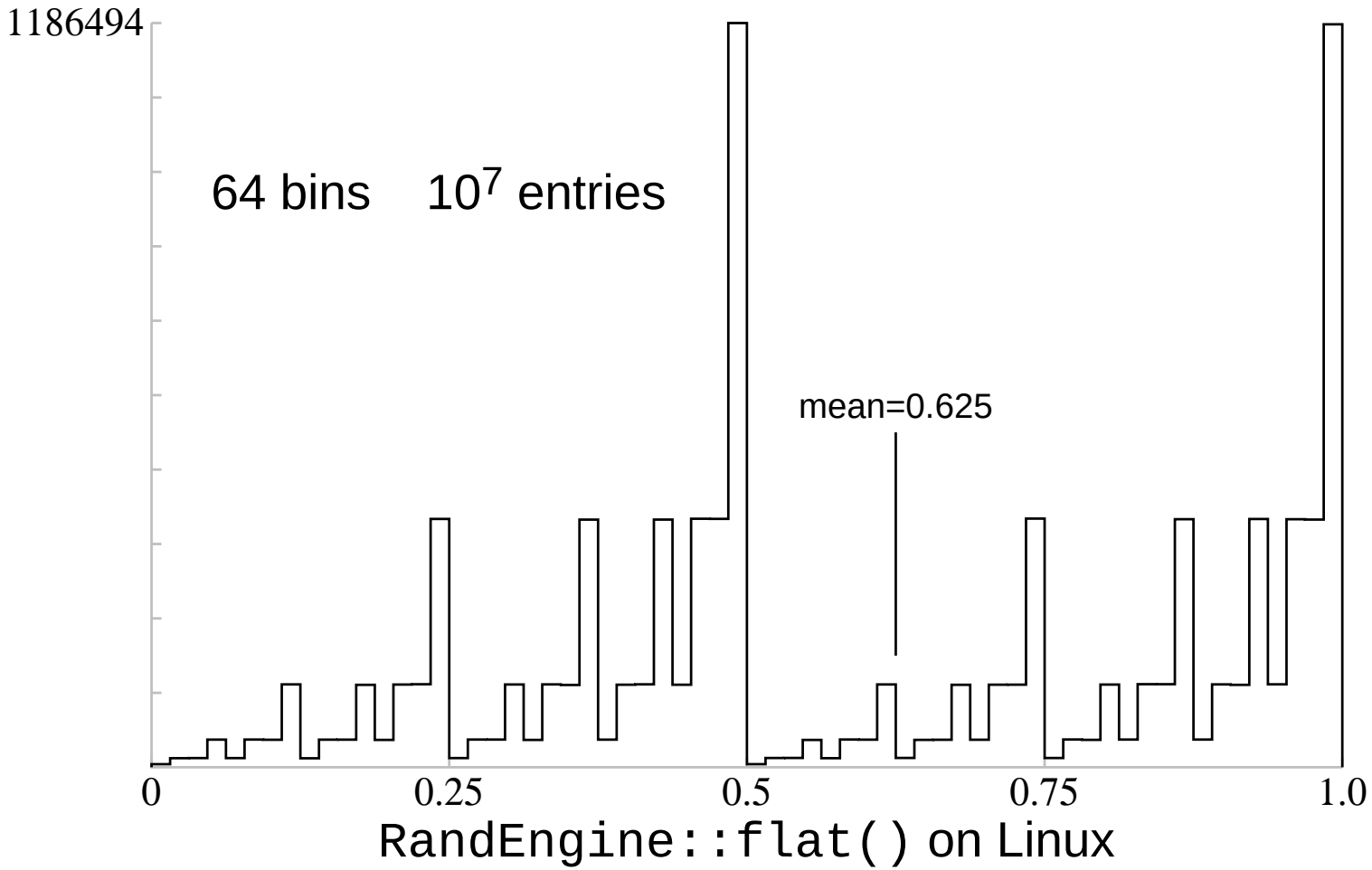
It would seem that the adoption of the `random` algorithm for C/C++ `rand` is a failed attempt to provide a "better" `rand` for Linux than is typically found in UNIX. A "standard" method (e.g. `ranlux`) should have been adopted instead. The defect in `random` is relatively minor, but there is no reason to use any generator that has any known statistical defect—you never know when the defect will bite you.

`rand` on UNIX (i.e. SGI and Sun) is only 15-bit, which is not sufficient for serious use. And, quoting NR in C:

> . . . its implementation in many, if not most, ANSI C libraries is quite flawed; quite a number of implementations are in the category "totally botched".

The fact that the C/C++ rand algorithm can be different on every system is also an excellent reason not to use it—even if it seems to work OK in one platform, on another it may cause serious problems, making validation a nightmare.

CLHEP's `RandEngine` (which is based on C/C++ rand) provides an extreme example; while operating reasonably with a 15-bit rand, on Linux `RandEngine::flat()` produces the following:



It's actually fractal, with more structure at smaller scales.

The `RandEngine` bug is due to the assumption in its implementation that `rand` is *always* a 15-bit generator. Obviously, `RandEngine` can't have been used extensively in the past on Linux, since a bug of this magnitude would have been noticed eventually, but a search in the CDF code browser does give a few hits. . .

In conclusion, while I'm sure CLHEP's `RandEngine` will be fixed promptly, the real underlying problem remains the fact that the C/C++ standard leaves the choice of the algorithm for `rand` to system implementors, who have a long history of providing bad generators. Quoting from NR again:

> If all scientific papers whose results are in doubt because of bad `rand()`'s were to disappear from library shelves, there would be a gap on each shelf about as big as your fist.